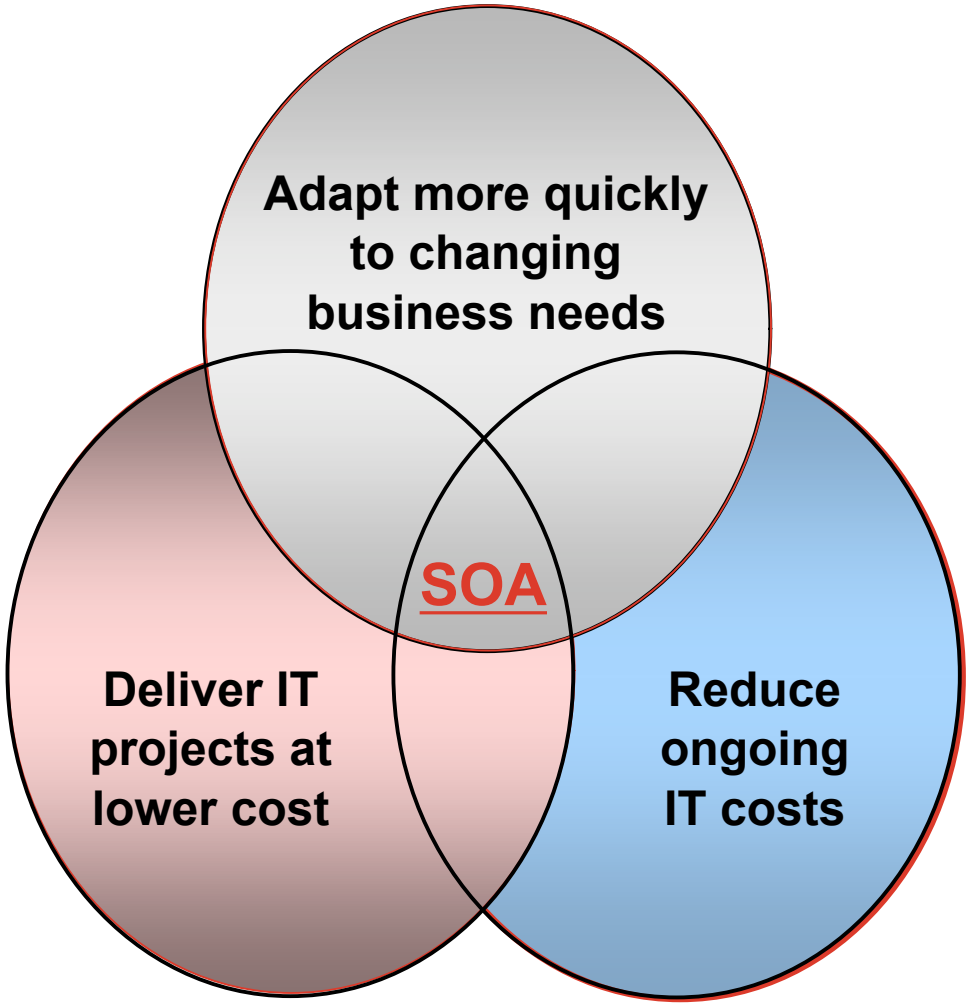
Actional
SOA COMMAND AND CONTROL

Creating a Successful SOA Initiative

Daniel M. Foody
CTO

http://www.actional.com
dan@actional.com

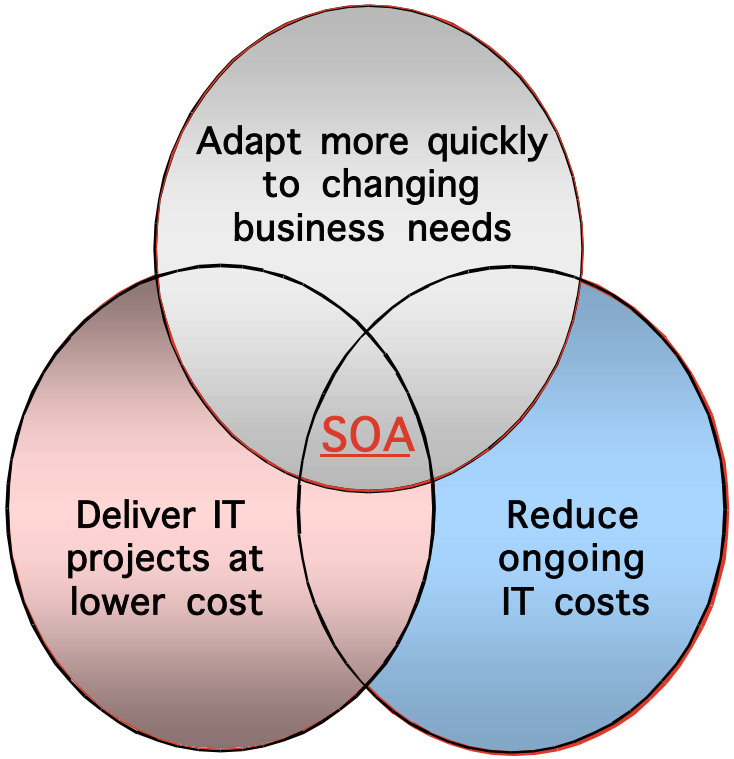
The benefits of SOA are compelling



Actional

Architecture initiatives that deliver SOA benefits

Decoupling



Integration

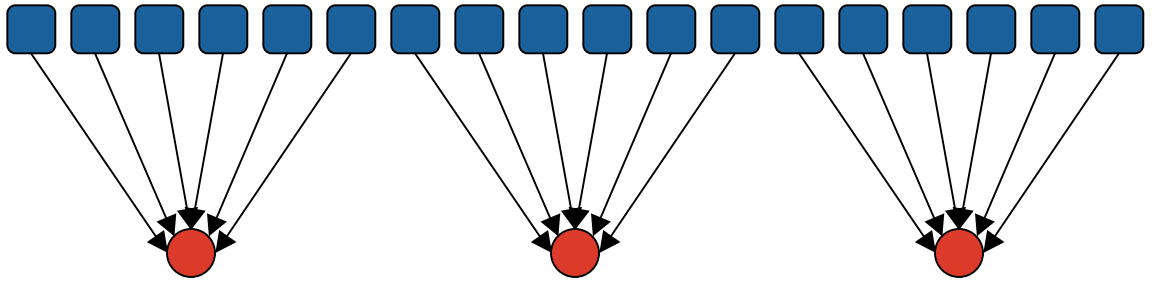
Reuse

3 fundamental tenets of successful SOA initiatives

- Quantify
  - » Business benefit of SOA initiatives must be measured
    - Not just measurable
  - » Accountability for roles must be unambiguous
- Qualify
  - » Select pilots for SOA initiative carefully, to assure success
  - » Not all services are created equal
    - Apply the right set of rules to the right services
- Align
  - » The interests of project teams must be aligned with the interests of the SOA initiative

Quantifying SOA
Which SOA initiative has been more successful?

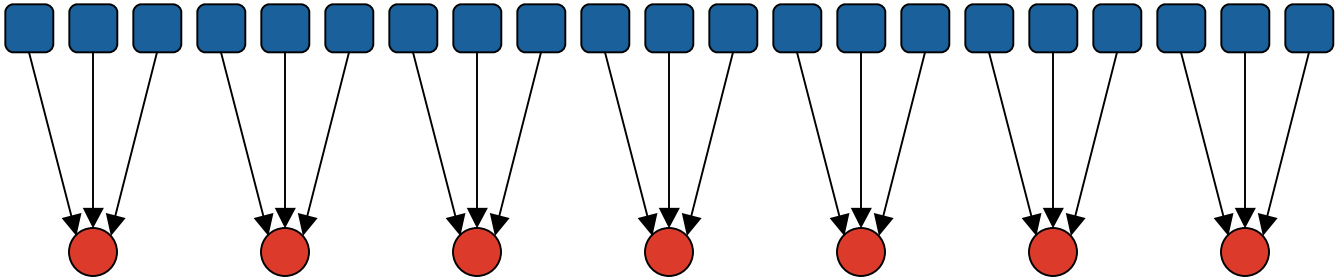
SOA Initiative #1



18 Consumer apps

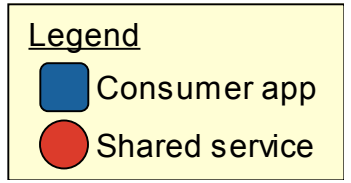
3 Shared services

SOA Initiative #2



21 Consumer apps

7 Shared services



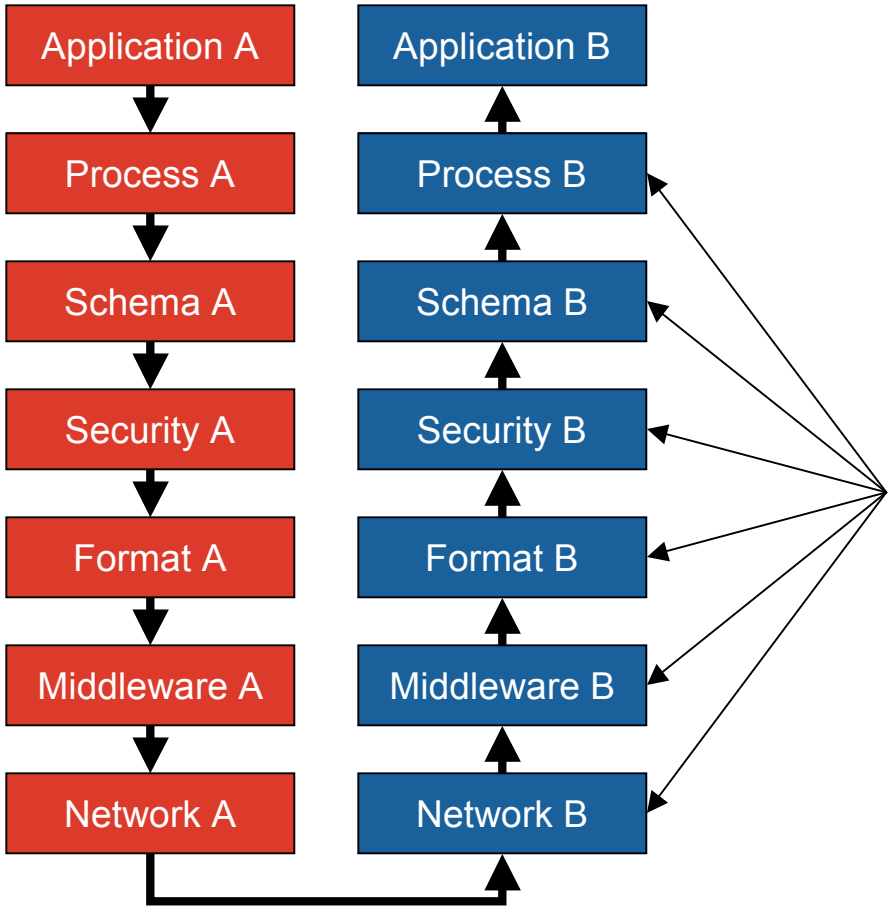
Quantifying SOA
Objective measurements

- Measurement principles:
  - » Your organization should serve the same number of business needs whether you adopt SOA or not (i.e. number of "consumers" same)
  - » Cost to build, maintain, operate, and use a service can be determined
    - Whether shared or single-purpose

- Purpose of SOA measurements:
  - » **Number of *re*uses of shared services** measures the effectiveness of SOA
    - Each reuse results in the cost avoidance or reduction of building, maintaining, and operating a single-purpose service
  - » **Number of *shared* service consumers** (relative to total consumers) measures the breadth of SOA adoption in your organization
    - Mostly a measure of how well the "cultural shift" of SOA has permeated the organization – not directly correlated with SOA business benefit
  - » **Number of services** measures the adoption of web services
    - Not directly meaningful from a business context

Actional

SOA integration initiative:
The barriers to integrating applications



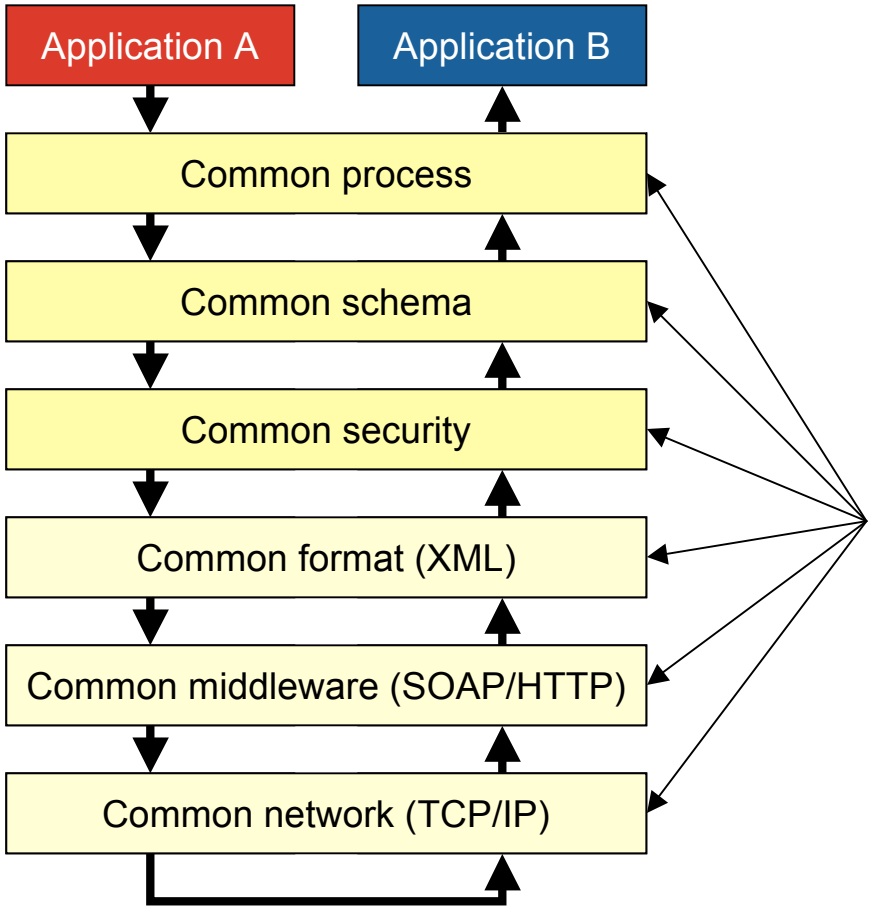
Integration has traditionally been an afterthought

Each layer that needs to be adapted adds cost, complexity, and risk

Specialized tools needed to handle all of these differences

Actional

SOA integration initiative:
Increasing the commonality between applications



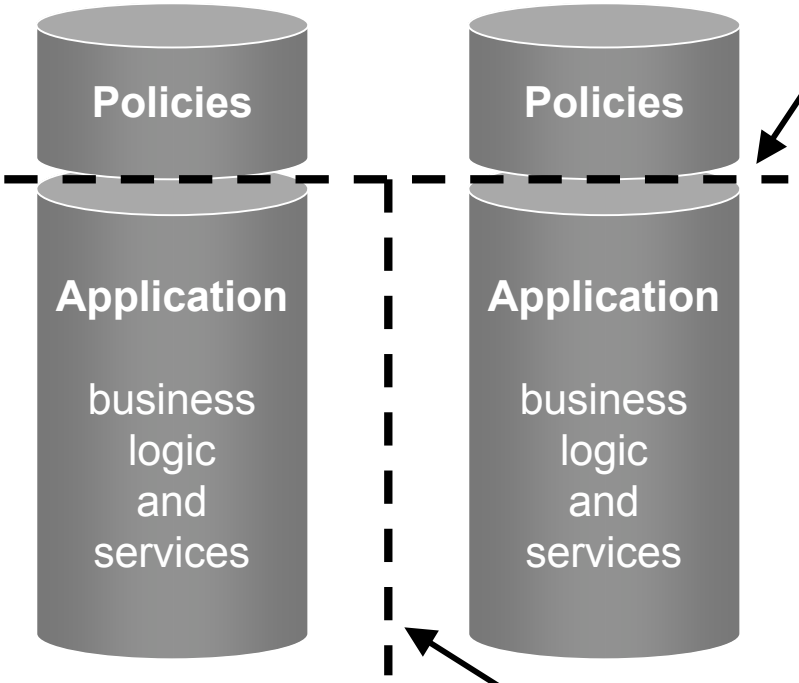
Build-in the ability to integrate the application from day-one

Choose common standards to underpin new applications – go as far up the stack as possible

With common standards, cost and risk are dramatically reduced, the need for specialized tools is greatly diminished

Actional

SOA decoupling initiative:
Proactively managing the impact of change



Policy decoupling

(compliance, entitlement, security, SLA, …)

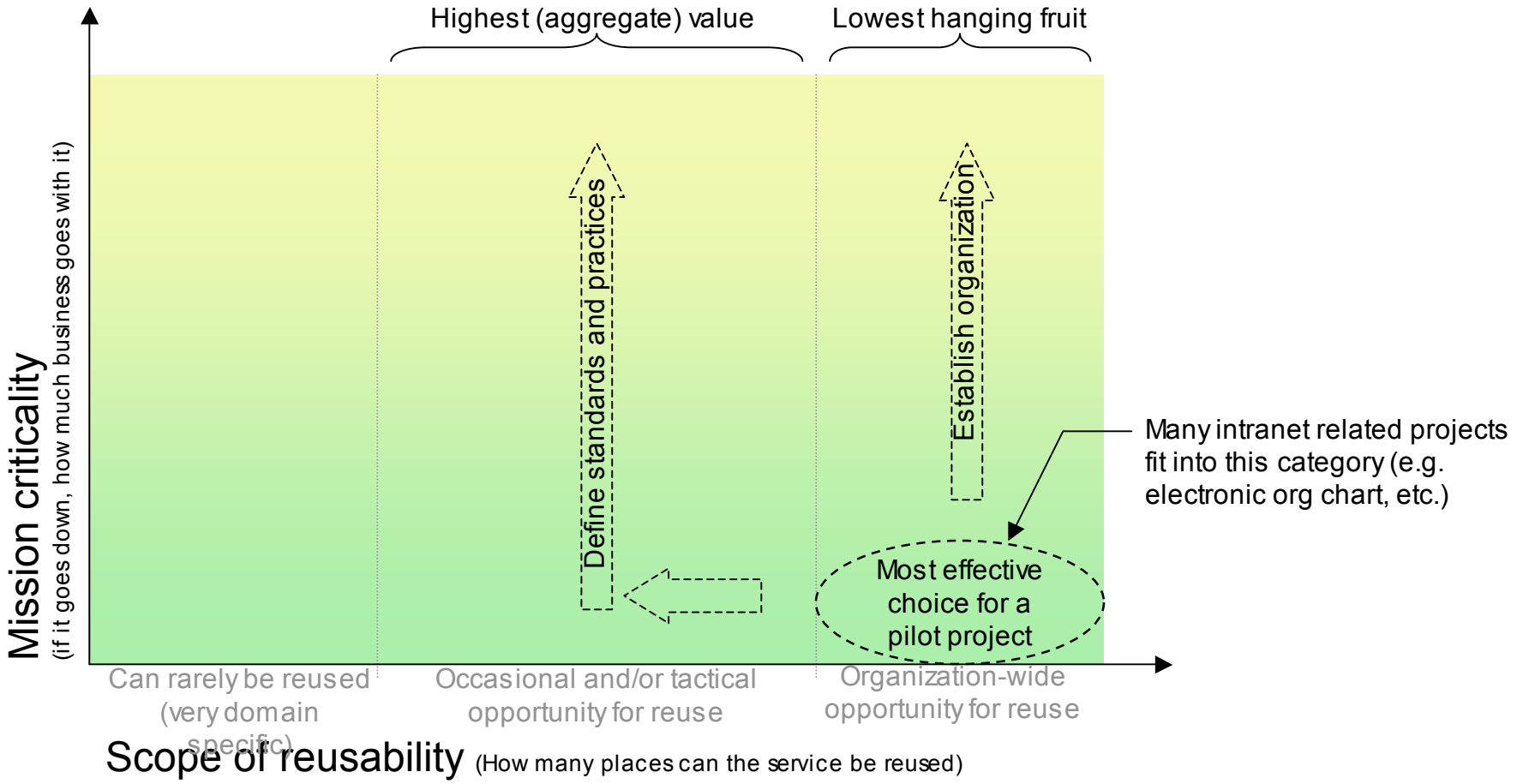
Minimize the impact that change in policies can have on applications

Application decoupling

(version upgrade, load increases, outages, …)

Control and minimize the impact that change in one application can have on another

Qualifying services



Qualifying services

- Organization-wide opportunity for reuse
  - ✓ Cost justification exists to dedicate teams/projects to building and maintaining these services
  - ⚠ Organization-wide shared services implicitly defines common standards for
    - Process, schema, security, etc.
- Occasional and/or tactical opportunity for reuse
  - ✓ Infrastructure, standards, and practices should be in-place to make sure that services have reasonably likelihood of reusability
  - ⚠ It's not normal for teams to want to share and reuse

Aligning interests

⚠ Short-term benefits (e.g. faster project delivery) and long-term benefits (e.g. lower cost of ownership) are often in conflict

✓ Choose “dual use” application infrastructure

» Offloads project teams from “grunt” infrastructure work

- Security, routing, load management, transformation, monitoring, logging, exception management, QoS, versioning, etc.

▶ Makes delivering the projects quicker, easier, less risky

» Automatically uses/enforces the corporate standards

- Security, schemas, industry standards, directory, etc.

▶ Sets the services up for having low integration cost when reused

Aligning interests

- Shouldn't make sharing and reuse a burden for project teams
  - » e.g. draconian rules for testing, scalability, etc.
  - » You won't know many of the ways a service can be reused *in advance*
    - Forcing design for hypothetical re-use cases is the recipe for building a kitchen sink

Uncontrolled reuse can cause a service to be stretched beyond its original design limits

- » Causing unintended failures and problems

✓ Use "dual-use" infrastructure to provide policy-controlled reuse

- » Anyone can reuse a service (but always through infrastructure layer)
- » Infrastructure will automatically throttle load on the service
  - Keeping it in a range that results in low-usage
- » To use service beyond this level, service must undergo a formal certification process (e.g. including load testing)
  - Infrastructure-enforced bottleneck is relaxed as service is formally certified at different load levels

Aligning interests

Don't force sharing and/or reuse if it would materially impact a project

- » A service built for the use of one team may not be suitable for a every subsequent use

- Redundancy is *normal* and *natural* in an SOA!

✓ Formalize a process to control redundancy *via consolidation*

- » Once redundancy reaches a prescribed "severity" level for a given logical service (e.g. 3 redundant implementations) trigger the consolidation process
  - May include launching a project to pick "winning" service implementation and decommission others

For more information on:

- Products to make your SOA and Web services secure, reliable and manageable
- Services to enable a successful migration to SOA in *your* enterprise

Visit http://www.actional.com